

# Improved constructions of quantum automata

Andris Ambainis and Nikolajs Nahimovs\*

Department of Computer Science, University of Latvia, Raina bulv. 19, Riga,  
LV-1586, Latvia, [andris.ambainis@lu.lv](mailto:andris.ambainis@lu.lv), [kolja.nahimov@gmail.com](mailto:kolja.nahimov@gmail.com).

**Abstract.** We present a simple construction of quantum automata which achieve an exponential advantage over classical finite automata. Our automata use  $\frac{4}{\epsilon} \log 2p + O(1)$  states to recognize a language that requires  $p$  states classically. The construction is both substantially simpler and achieves a better constant in the front of  $\log p$  than the previously known construction of [2].

Similarly to [2], our construction is by a probabilistic argument. We consider the possibility to derandomize it and present some results in this direction.

## 1 Introduction

Quantum finite automata are a mathematical model for quantum computers with limited memory. A quantum finite automaton has a finite state space and applies a sequence of transformations, corresponding to the letter of the input word to this state space. At the end, the state of the quantum automaton is measured and the input word is accepted or rejected, depending on the outcome of the measurement.

Most commonly, finite automata (including quantum finite automata) are studied in 1-way model where the transformations corresponding to the letters of the input word are applied in the order of the letters in the word, from the left to the right. (More general 2-way models [8] allow the order of the transformations to depend on the results of the previous transformations.)

For 1-way model (which we consider the most natural model in the quantum setting), the set of languages (computational problems) that can be recognized (computed) by a quantum automaton is the same for classical automata<sup>1</sup>. However, quantum automata can be exponentially more space-efficient than classical automata [2]. This is one of only two results that show an exponential advantage for quantum algorithms in space complexity. (The other is the recent exponential separation for online algorithms by Le Gall [9].)

Our first result is an improved exponential separation between quantum and classical finite automata, for the same computational problem as in

---

\* Supported by University of Latvia research project Y2-ZP01-100.

<sup>1</sup> More precisely, this is true for sufficiently general models of quantum automata, such as one proposed in [5] or [7]. There are several results claiming that quantum automata are weaker than classical (e.g. [8,3,4]) but this is an artifact of restrictive models of quantum automata being used.

[2]. The construction in [2] is quite inefficient. While it produces an example where classical automata require  $p$  states and quantum automata require  $C \log p$  states, the constant  $C$  is fairly large. In this paper, we provide a new construction with a better constant and, also, a much simpler analysis. (A detailed comparison between our results and [2] is given in section 3.1.)

Second, both construction of QFAs in [2] and this paper are probabilistic. That is, they employ a sequence of parameters that are chosen at random and hardwired into the QFA. In the last section, we give two non-probabilistic constructions of QFAs for the same language. The first of them gives QFAs with  $O(\log p)$  states but its correctness is only shown by numerical experiments. The second construction gives QFAs with  $O(\log^{2+\epsilon} p)$  states but is provably correct.

## 2 Definitions

### 2.1 Quantum finite automata

We consider 1-way quantum finite automata (QFA) as defined in [10]. Namely, a 1-way QFA is a tuple  $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$  where  $Q$  is a finite set of states,  $\Sigma$  is an input alphabet,  $\delta$  is a transition function,  $q_0 \in Q$  is a starting state,  $Q_{acc}$  and  $Q_{rej}$  are sets of accepting and rejecting states and  $Q = Q_{acc} \cup Q_{rej}$ .  $\phi$  and  $\$$  are symbols that do not belong to  $\Sigma$ . We use  $\phi$  and  $\$$  as the left and the right endmarker, respectively. The *working alphabet* of  $M$  is  $\Gamma = \Sigma \cup \{\phi, \$\}$ .

A superposition of  $M$  is any element of  $l_2(Q)$  (the space of mappings from  $Q$  to  $\mathbb{C}$  with  $l_2$  norm). For  $q \in Q$ ,  $|q\rangle$  denotes the unit vector with value 1 at  $q$  and 0 elsewhere. All elements of  $l_2(Q)$  can be expressed as linear combinations of vectors  $|q\rangle$ . We will use  $\psi$  to denote elements of  $l_2(Q)$ .

The transition function  $\delta$  maps  $Q \times \Gamma \times Q$  to  $\mathbb{C}$ . The value  $\delta(q_1, a, q_2)$  is the amplitude of  $|q_2\rangle$  in the superposition of states to which  $M$  goes from  $|q_1\rangle$  after reading  $a$ . For  $a \in \Gamma$ ,  $V_a$  is a linear transformation on  $l_2(Q)$  defined by

$$V_a(|q_1\rangle) = \sum_{q_2 \in Q} \delta(q_1, a, q_2) |q_2\rangle. \quad (1)$$

We require all  $V_a$  to be unitary.

The computation of a QFA starts in the superposition  $|q_0\rangle$ . Then transformations corresponding to the left endmarker  $\phi$ , the letters of the input word  $x$  and the right endmarker  $\$$  are applied. The transformation corresponding to  $a \in \Gamma$  is just  $V_a$ . If the superposition before reading  $a$  is  $\psi$ , then the superposition after reading  $a$  is  $V_a(\psi)$ .

After reading the right endmarker, the current state  $\psi$  is observed with respect to the observable  $E_{acc} \oplus E_{rej}$  where  $E_{acc} = \text{span}\{|q\rangle : q \in Q_{acc}\}$ ,  $E_{rej} = \text{span}\{|q\rangle : q \in Q_{rej}\}$ . This observation gives  $x \in E_i$  with the probability equal to the square of the projection of  $\psi$  to  $E_i$ . After that, the superposition collapses to this projection.

If we get  $\psi \in E_{acc}$ , the input is accepted. If  $\psi \in E_{rej}$ , the input is rejected.

**Another definition of QFAs.** Independently of [10], quantum automata were introduced in [8]. There is one difference between these two definitions. In [8], a QFA is observed after reading each letter (after doing each  $V_a$ ). In [10], a QFA is observed only after all letters have been read. The definition of [8] is more general. But, in this paper, we follow the definition of [10] because it is simpler and sufficient to describe our automaton.

## 2.2 Unitary transformations

We use the following theorem from linear algebra.

**Theorem 1.** *Let  $\alpha_1, \dots, \alpha_m$  be such that  $|\alpha_1|^2 + \dots + |\alpha_m|^2 = 1$ . Then,*

1. *there is a unitary transformation  $U_1$  such that  $U_1|q_1\rangle = \alpha_1|q_1\rangle + \dots + \alpha_m|q_m\rangle$ .*
2. *there is a unitary transformation  $U_2$  such that, for all  $i \in \{1, \dots, m\}$ ,  $U_2|q_i\rangle$  is equal to  $\alpha_i|q_1\rangle$  plus some combination of  $|q_2\rangle, \dots, |q_m\rangle$ .*

In the second case, we also have

$$U_2(\alpha_1|q_1\rangle + \dots + \alpha_m|q_m\rangle) = |q_1\rangle.$$

## 3 Space-efficient quantum automaton

### 3.1 Summary of results

Let  $p$  be a prime. We consider the language  $L_p = \{ a^i \mid i \text{ is divisible by } p \}$ . It is easy to see that any deterministic 1-way finite automaton recognizing  $L_p$  has at least  $p$  states. However, there is a much more efficient QFA! Namely, Ambainis and Freivalds [2] have shown that  $L_p$  can be recognized by a QFA with  $O(\log p)$  states.

The big-O constant in this result depends on the required probability of correct answer. For  $x \in L_p$ , the answer is always correct with probability 1. For  $x \notin L_p$ , [2] give

- a QFA with  $16 \log p$  states that is correct with probability at least  $1/8$  on inputs  $x \notin L_p$ .
- a QFA with  $\text{poly}(\frac{1}{\epsilon}) \log p$  states that is correct with probability at least  $1 - \epsilon$  on inputs  $x \notin L_p$  (where  $\text{poly}(x)$  is some polynomial in  $x$ ).

In this paper, we present a simpler construction of QFAs that achieves a better big-O constant.

**Theorem 2.** *For any  $\epsilon > 0$ , there is a QFA with  $4 \frac{\log 2p}{\epsilon}$  states recognizing  $L_p$  with probability at least  $1 - \epsilon$ .*

### 3.2 Proof of Theorem 2

Let  $U_k$ , for  $k \in \{1, \dots, p-1\}$ , be a quantum automaton with a set of states  $Q = \{q_0, q_1\}$ , a starting state  $|q_0\rangle$ ,  $Q_{acc} = \{q_0\}$ ,  $Q_{rej} = \{q_1\}$ . The transition function is defined as follows. Reading  $a$  maps  $|q_0\rangle$  to  $\cos \phi |q_0\rangle + \sin \phi |q_1\rangle$  and  $|q_1\rangle$  to  $-\sin \phi |q_0\rangle + \cos \phi |q_1\rangle$  where  $\phi = \frac{2\pi k}{p}$ . (It is easy to check that this transformation is unitary.) Reading  $\$$  and  $\$$  leaves  $|q_0\rangle$  and  $|q_1\rangle$  unchanged.

**Lemma 1.** *After reading  $a^j$ , the state of  $U_k$  is*

$$\cos\left(\frac{2\pi jk}{p}\right)|q_0\rangle + \sin\left(\frac{2\pi jk}{p}\right)|q_1\rangle.$$

**Proof.** By induction.  $\square$

If  $j$  is divisible by  $p$ , then  $\frac{2\pi jk}{p}$  is a multiple of  $2\pi$ ,  $\cos(\frac{2\pi jk}{p}) = 1$ ,  $\sin(\frac{2\pi jk}{p}) = 0$ , reading  $a^j$  maps the starting state  $|q_0\rangle$  to  $|q_0\rangle$ . Therefore, we get an accepting state with probability 1. This means that all automata  $U_k$  accept words in  $L$  with probability 1.

Let  $k_1, \dots, k_d$  be a sequence of  $d = c \log p$  numbers. We construct an automaton  $U$  by combining  $U_{k_1}, \dots, U_{k_d}$ . The set of states consists of  $2d$  states  $q_{1,0}, q_{1,1}, q_{2,0}, q_{2,1}, \dots, q_{d,0}, q_{d,1}$ . The starting state is  $q_{1,0}$ .

The transformation for left endmarker  $\$$  is such that  $V_{\$}(|q_{1,0}\rangle) = |\psi_0\rangle$  where

$$|\psi_0\rangle = \frac{1}{\sqrt{d}}(|q_{1,0}\rangle + |q_{2,0}\rangle + \dots + |q_{d,0}\rangle).$$

This transformation exists by first part of Theorem 1. The transformation for  $a$  is defined by

$$V_a(|q_{i,0}\rangle) = \cos \frac{2k_i \pi}{p} |q_{i,0}\rangle + \sin \frac{2k_i \pi}{p} |q_{i,1}\rangle,$$

$$V_a(|q_{i,1}\rangle) = -\sin \frac{2k_i \pi}{p} |q_{i,0}\rangle + \cos \frac{2k_i \pi}{p} |q_{i,1}\rangle.$$

The transformation  $V_{\$}$  is as follows. The states  $|q_{i,1}\rangle$  are left unchanged. On the states  $|q_{i,0}\rangle$ ,  $V_{\$}|q_{i,0}\rangle$  is  $\frac{1}{\sqrt{d}}|q_{1,0}\rangle$  plus some other state (part 2 of Theorem 1, applied to  $|q_{1,0}\rangle, \dots, |q_{d,0}\rangle$ ). In particular,

$$V_{\$}|\psi_0\rangle = |q_{1,0}\rangle.$$

The set of accepting states  $Q_{acc}$  consists of one state  $q_{1,0}$ . All other states  $q_{i,j}$  belong to  $Q_{rej}$ .

*Claim.* If the input word is  $a^j$  and  $j$  is divisible by  $p$ , then  $U$  accepts with probability 1.

*Proof.* The left endmarker maps the starting state to  $|\psi_0\rangle$ . Reading  $j$  letters  $a$  maps each  $|q_{i,0}\rangle$  to itself (see analysis of  $U_k$ ). Therefore, the state  $|\psi_0\rangle$  which consists of various  $|q_{i,0}\rangle$  is also mapped to itself. The right endmarker maps  $|\psi_0\rangle$  to  $|q_{1,0}\rangle$  which is an accepting state.  $\square$

*Claim.* If the input word is  $a^j$ ,  $j$  not divisible by  $p$ ,  $U$  accepts with probability

$$\frac{1}{d^2} \left( \cos \frac{2\pi k_1 j}{p} + \cos \frac{2\pi k_2 j}{p} + \dots + \cos \frac{2\pi k_d j}{p} \right)^2. \quad (2)$$

*Proof.* By Lemma 1,  $a^j$  maps  $|q_{i,0}\rangle$  to  $\cos \frac{2\pi k_i j}{p} |q_{i,0}\rangle + \sin \frac{2\pi k_i j}{p} |q_{i,1}\rangle$ . Therefore, the state before reading the right endmarker  $\$$  is

$$\frac{1}{\sqrt{d}} \sum_{i=1}^d \left( \cos \frac{2\pi k_i j}{p} |q_{i,0}\rangle + \sin \frac{2\pi k_i j}{p} |q_{i,1}\rangle \right).$$

The right endmarker maps each  $|q_{i,0}\rangle$  to  $\frac{1}{\sqrt{d}} |q_{1,0}\rangle$  plus superposition of other basis states. Therefore, the state after reading the right endmarker  $\$$  is

$$\frac{1}{d} \sum_{i=1}^d \cos \frac{2\pi k_i j}{p} |q_{1,0}\rangle$$

plus other states  $|q_{i,j}\rangle$ . Since  $|q_{1,0}\rangle$  is the only accepting state, the probability of accepting is the square of the coefficient of  $|q_{1,0}\rangle$ . This proves the lemma.  $\square$

We use the following theorem from probability theory (variant of Azuma's theorem[11]).

**Theorem 3.** Let  $X_1, \dots, X_d$  be independent random variables such that  $E[X_i] = 0$  and the value of  $X_i$  is always between -1 and 1. Then,

$$\Pr \left[ \left| \sum_{i=1}^d X_i \right| \geq \lambda \right] \leq 2e^{-\frac{\lambda^2}{2d}}.$$

We apply this theorem as follows. Fix  $j \in \{1, \dots, p-1\}$ . Pick each of  $k_1, \dots, k_d$  randomly from  $\{0, \dots, p-1\}$ . Define  $X_i = \cos \frac{2\pi k_i j}{p}$ . We claim that  $X_i$  satisfy the conditions of theorem. Obviously, the value of cos function is between -1 and 1. The expectation of  $X_i$  is

$$E[X_i] = \frac{1}{p} \sum_{k=0}^{p-1} \cos \frac{2\pi k j}{p}$$

since  $k_i = k$  for each  $k \in \{0, \dots, p-1\}$  with probability  $1/p$ . We have  $\cos \frac{2\pi k j}{p} = \cos \frac{2\pi (kj \bmod p)}{p}$  because  $\cos(2\pi + x) = \cos x$ . Consider the numbers  $0, j, 2j \bmod p, \dots, (p-1)j \bmod p$ . They are all distinct. (Since  $p$  is prime,  $kj = k'j \pmod p$  implies  $k = k'$ .) Therefore, the numbers  $0, j, 2j \bmod p, \dots, (p-1)j \bmod p$  are just  $0, 1, \dots, p-1$  in a different order. This means that the expectation of  $X_i$  is

$$E[X_i] = \frac{1}{p} \sum_{k=0}^{p-1} \cos \frac{2\pi k}{p}.$$

This is equal to 0.

By equation (2), the probability of accepting  $a^j$  is  $\frac{1}{d^2}(X_1 + \dots + X_d)^2$ . To achieve

$$\frac{1}{d^2}(X_1 + \dots + X_d)^2 \leq \epsilon,$$

we need  $|X_1 + \dots + X_d| \leq \sqrt{\epsilon}d$ . By Theorem 3, the probability that this does not happen is at most  $2e^{-\frac{\epsilon d}{2}}$ .

There are  $p - 1$  possible inputs not in  $L$ :  $a^1, \dots, a^{p-1}$ . The probability that one of them gets accepted with probability more than  $\epsilon$  is at most  $2(p-1)e^{-\frac{\epsilon d}{2}}$ . If

$$2(p-1)e^{-\frac{\epsilon d}{2}} < 1, \quad (3)$$

then there is at least one choice of  $k_1, \dots, k_d$  for which  $U$  does not accept any of  $a^1, \dots, a^{p-1}$  with probability more than  $\epsilon$ . The equation (3) is true if we take  $d = 2\frac{\log 2p}{\epsilon}$ . The number of states for  $U$  is  $4\frac{\log 2p}{\epsilon}$ .  $\square$

## 4 Explicit constructions of QFAs

In the previous section, we proved what for every  $\epsilon > 0$  and  $p \in P$ , there is a QFA with  $4\frac{\log 2p}{\epsilon}$  states recognizing  $L_p$  with probability at least  $1 - \epsilon$ . The proposed QFA construction depends on  $d = 2\frac{\log 2p}{\epsilon}$  parameters  $k_1, \dots, k_d$  and accepts input word  $a^j \notin L_p$  with probability

$$\frac{1}{d^2} \left( \sum_{i=1}^d \cos \frac{2\pi k_i j}{p} \right)^2.$$

It is possible to choose  $k_1, \dots, k_d$  values to ensure

$$\frac{1}{d^2} \left( \sum_{i=1}^d \cos \frac{2\pi k_i j}{p} \right)^2 < \epsilon$$

or, equivalently,

$$\left| \sum_{i=1}^d \cos \frac{2\pi k_i j}{p} \right| < \sqrt{\epsilon}d \quad (4)$$

for every  $a^j \notin L_p$ .

However, our proof is by a probabilistic argument and does not give an explicit sequence  $k_1, \dots, k_d$ . We now present two constructions of explicit sequences. The first construction works well in numerical experiments and gives a QFA with  $O(\log p)$  states in all the cases that we tested. The second construction uses a slightly larger number of states but has a rigorous proof of correctness.

### 4.1 The first construction: cyclic sequences

We conjecture

**Hypothesis 1** *If  $g$  is a primitive root modulo  $p \in P$ , then sequence  $S_g = \{k_i \equiv g^i \pmod{p}\}_{i=1}^d$  for all  $d$  and all  $j : a^j \notin L_p$  satisfies (4).*

We will call  $g$  a *sequence generator*. The corresponding sequence will be referred as cyclic sequence. We have checked all  $p \in \{2, \dots, 9973\}$ , all generators  $g$  and all sequence lengths  $d < p$  (choosing a corresponding  $\epsilon$  value) and haven't found any counterexample to our hypothesis.

We now describe numerical experiments comparing two strategies: using a random sequence  $k_1, \dots, k_d$  and using a cyclic sequence.

We will use  $S_{rand}$  to denote random sequence and  $S_g$  to denote a cyclic sequence with generator  $g$ . We will also use  $\epsilon_{rand}$  and  $\epsilon_g$  to denote the maximum probability with which a corresponding automata accepts input word  $a^j \notin L_p$ .

Table 1 shows  $\epsilon_{rand}$  and  $\epsilon_g$  for different  $p$  and  $g$  values.  $\epsilon_{rand}$  is calculated as an average over 5000 randomly selected sequences.  $\epsilon_g$  is for one specific generator.  $\epsilon$  in the second column shows the theoretical upper bound given by Theorem 2.

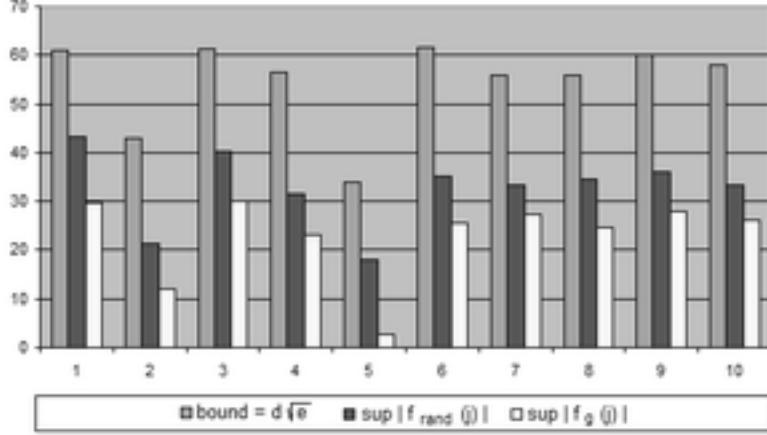
| $p$  | $\epsilon$ | $d$ | $g$  | $\epsilon_{rand}$ | $\epsilon_g$ |
|------|------------|-----|------|-------------------|--------------|
| 1523 | 0,1        | 161 | 948  | 0,03635           | 0,01517      |
| 2689 | 0,1        | 172 | 656  | 0,03767           | 0,01950      |
| 3671 | 0,1        | 179 | 2134 | 0,03803           | 0,02122      |
| 4093 | 0,1        | 181 | 772  | 0,03822           | 0,01803      |
| 5861 | 0,1        | 188 | 2190 | 0,03898           | 0,01825      |
| 6247 | 0,1        | 189 | 406  | 0,03922           | 0,02006      |
| 7481 | 0,1        | 193 | 6978 | 0,03932           | 0,01691      |
| 8581 | 0,1        | 196 | 5567 | 0,03942           | 0,02057      |
| 9883 | 0,1        | 198 | 1260 | 0,04011           | 0,01905      |

**Table 1.**  $\epsilon_{rand}$  and  $\epsilon_g$  for different  $p$  and  $g$

In 99.98% - 99.99% of our experiments, random sequences achieved the bound of Theorem 2. Surprisingly, cyclic sequences substantially outperform random ones in almost all the cases.

More precisely, for randomly selected  $p \in P$ ,  $\epsilon > 0$  and generator  $g$ , a cyclic sequence  $S_g$  gives a better result than a random sequence  $S_{rand}$  in 98.29% of cases. A few random instances are shown in Figure 1. For each instance, we show the bound  $d\sqrt{\epsilon}$  on (4) obtained by a probabilistic argument, the maximum of  $f_{rand}(j)$  (which is defined as the value of (4) for the sequence  $S_{rand}$ ) over all  $j$ ,  $a^j \notin L_p$  and the maximum of  $f_g(j)$  (defined in a similar way using  $S_g$  instead of  $S_{rand}$ ).

In 1.81% of cases, we got that  $\sup |f_g(j)| > \sup |f_{rand}(j)|$ , where  $\sup |f_{rand}(j)|$  is calculated as an average over 5000 randomly selected sequences. Figure 2 shows one of these cases:  $p = 9059$ ,  $\epsilon = 0.09$  and  $g = 2689$ , comparing the cyclic sequence with 9 different randomly chosen sequences. The cyclic sequence gives a slightly worse result than most of the random ones, but still beats the probabilistic bound on (4) by a substantial amount.



**Fig. 1.**  $\sup |f_g(j)|$  and  $\sup |f_{rand}(j)|$  for random  $p, \epsilon$  and  $g$

**Comparing different generators** Every  $p \in P$  might have multiple generators. Table 2 shows  $\epsilon_g$  values for  $p = 9059$  and  $\epsilon = 0.1$  (sequence length  $d = 197$ ,  $\sqrt{\epsilon}d = 62.0101221453601$ ).

| $g$ | $\epsilon_g$ | $g$  | $\epsilon_g$ | $g$  | $\epsilon_g$ |
|-----|--------------|------|--------------|------|--------------|
| 102 | 0,02533      | 1545 | 0,01858      | 9023 | 0,01807      |
| 103 | 0,03758      | 1546 | 0,02235      | 9033 | 0,01413      |
| 105 | 0,01999      | 1549 | 0,02896      | 9034 | 0,01485      |
| 106 | 0,02852      | 1552 | 0,02873      | 9036 | 0,02509      |
| 110 | 0,01685      | 1553 | 0,02624      | 9039 | 0,02311      |

**Table 2.**  $\epsilon_g$  values for different generators.  $p = 9059$

Different generators have different  $\epsilon_g$  values. We will use  $g_{min}$  to refer a minimal generator, i.e. one having a minimal  $\epsilon_g$ . Table 3 shows minimal generators for  $p$  values from table 1.

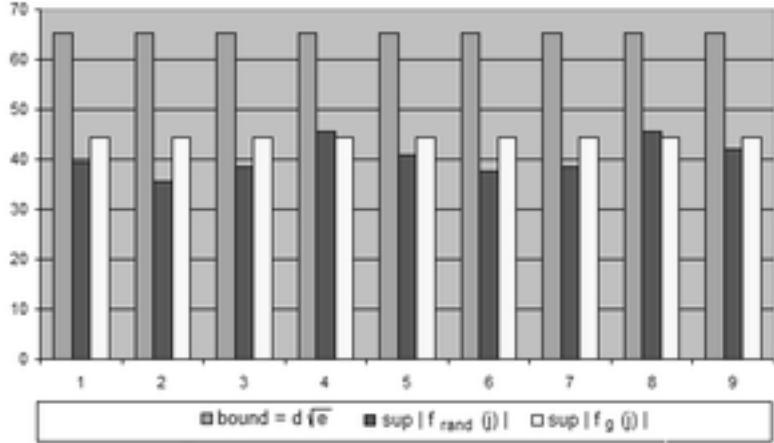
We see that, typically, the minimal generators give a QFA with substantially smaller probability of error. It remains open whether one could find a minimal generator without an exhaustive search of all generators.

## 4.2 The second construction: AIKPS sequences

Fix  $\epsilon > 0$ . Let

$$P = \{r \mid r \text{ is prime, } (\log p)^{1+\epsilon}/2 < r \leq (\log p)^{1+\epsilon}\},$$

$$S = \{1, 2, \dots, (\log p)^{1+2\epsilon}\},$$



**Fig. 2.**  $\sup |f_g(j)|$  and  $\sup |f_{rand}(j)|$  for  $p = 9059$ ,  $\epsilon = 0.09$  and  $g = 2689$

| $p$  | $\epsilon$ | $d$ | $g$  | $\epsilon_g$ | $g_{min}$ | $\epsilon_{g_{min}}$ |
|------|------------|-----|------|--------------|-----------|----------------------|
| 1523 | 0,1        | 161 | 948  | 0,01517      | 624       | 0,00919              |
| 2689 | 0,1        | 172 | 656  | 0,01950      | 1088      | 0,01060              |
| 3671 | 0,1        | 179 | 2134 | 0,02122      | 1243      | 0,01121              |
| 4093 | 0,1        | 181 | 772  | 0,01803      | 1063      | 0,01154              |
| 5861 | 0,1        | 188 | 2190 | 0,01825      | 5732      | 0,01133              |
| 6247 | 0,1        | 189 | 406  | 0,02006      | 97        | 0,01182              |
| 7481 | 0,1        | 193 | 6978 | 0,01691      | 2865      | 0,01205              |
| 8581 | 0,1        | 196 | 5567 | 0,02057      | 4362      | 0,01335              |
| 9883 | 0,1        | 198 | 1260 | 0,01905      | 5675      | 0,01319              |

**Table 3.** Minimal generators for different  $p$

$$T = \{s \cdot r^{-1} | r \in R, s \in S\},$$

with  $r^{-1}$  being the inverse modulo  $p$ . Ajtai et al. [1] have shown

**Theorem 4.** [1] For all  $k \in \{1, \dots, p-1\}$ ,

$$|\sum_{t \in T} e^{2tk\pi i/p}| \leq (\log p)^{-\epsilon} |T|.$$

Razborov et al. [12] have shown that powers  $e^{2tk\pi i/p}$  satisfy even stronger uniformity conditions. We, however, only need Theorem 4.

By taking the real part of the left hand side, we get

$$\left| \sum_{t \in T} \cos\left(\frac{2tk\pi i}{p}\right) \right| \leq (\log p)^{-\epsilon} |T|.$$

Thus, taking our construction of QFAs and using elements of  $T$  as  $k_1, \dots, k_d$  gives an explicit construction of a QFA for our language with  $O(\log^{2+3\epsilon})$  states.

For our first, cyclic construction, the best provable result is by applying a bound on exponential sums by Bourgain [6]. That gives a QFA with  $O(p^{c/\log \log p})$  states which is weaker than both the numerical results and the rigorous construction in this section.

**Acknowledgment.** We thank Igor Shparlinski for pointing out [1] and [6] to us.

## References

1. M. Ajtai, H. Iwaniec, J. Komlos, J. Pintz, E. Szemerédi. Construction of a thin set with small Fourier coefficients, *Bulletin of the London Mathematical Society*, 22:583-590, 1990.
2. A. Ambainis, R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. *Proceedings of the 39th IEEE Conference on Foundations of Computer Science*, 332-341, 1998. Also quant-ph/9802062.
3. A. Ambainis, A. Kikusts, M. Valdats. On the Class of Languages Recognizable by 1-Way Quantum Finite Automata. *Proceedings of STACS'01*, Lecture Notes in Computer Science, 2010:75-86, 2001.
4. A. Ambainis, A. Nayak, A. Ta-Shma, U. Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM*, 49(4): 496-511 (2002)
5. A. Bertoni, C. Mereghetti, B. Palano. Quantum Computing: 1-Way Quantum Automata. *Developments in Language Theory 2003*, Lecture Notes in Computer Science, 2710:1-20.
6. J. Bourgain. Estimates on exponential sums related to Diffie-Hellman distributions. *Geometric and Functional Analysis*, 15:1-34, 2005.
7. M. Ciamarra. Quantum Reversibility and a New Model of Quantum Automaton. *Proceedings of FCT'01*, 2138:376-379, 2001.
8. A. Kondacs, J. Watrous, On the power of quantum finite state automata. *Proceedings of the 38th IEEE Conference on Foundations of Computer Science*, 66-75, 1997.
9. F. Le Gall. Exponential separation of quantum and classical online space complexity. *Proceedings of SPAA '06*, 67-73.
10. C. Moore, J. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237(1-2): 275-306 (2000). Also quant-ph/9707031.
11. R. Motwani, P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1994.
12. A. Razborov, E. Szemerédi, A. Wigderson, Constructing small sets that are uniform in arithmetic progressions. *Combinatorics, Probability and Computing*, 2:513-518, 1993.